

AMENDMENTS TO THE SPECIFICATION

Please delete paragraph [0019] at page 3 of the Specification.

Please replace paragraph [0028] with the following amended paragraph:

[0028] FIG. 2 illustrates a system embodiment 200 of the invention that may be implemented in one or more components of system 100. System 200 may include processor 202, PMU (performance monitor unit) 204, memory 206, performance monitor driver 208, compressor 210, event buffer ~~204~~ 218, and decompressor 214. System 200 is not limited to the components illustrated, and is not limited to the number of components illustrated. For example, system 200 may include a plurality of processors 202, and/or other components, without departing from embodiments of the invention.

Please replace paragraph [0028] with the following amended paragraph:

[0029] Processor 202 may be a processor, such as host processor 102. Processor 202 may include PMU 204 to monitor system 200 for one or more events, and to collect the one or more events in memory 206. An "event" as used herein, refers to acts associated with performance of a system, such as system 100, or system 200. For example, an event may comprise an instruction cache miss event, a data cache miss event, or a branch event. An instruction

cache miss event refers to the execution of a program instruction that may result, at least in part, in a miss in an instruction cache. A data cache miss event refers to a data access that may result, at least in part, in a cache miss. A branch event may record the outcome of the execution of a branch instruction. A sequence of consecutive branch events may be recorded by PMU 204 to, forming a program path profile, also known as (or a branch trace[]). Other types of events, of course, are possible. ~~However, these~~ These types of events will be further described and/or illustrated.

Please replace paragraph [0034] with the following amended paragraph:

[0034] At block 308, performance monitor driver 208 may store one or more event data in processed event record 216. The method ends at block 310. Performance monitor driver 208 may store processed event record 216 in event buffer 204 218. Client application may access one or more processed event records 216 from event buffer 204 218.

Please replace paragraph [0035] with the following amended paragraph:

[0035] PMU 204 and performance monitor driver 208 may each be embodied in machine-executable instructions, such as machine-executable instructions 130, 132 that may be executed by processor 202, such as host processor 102, and/or circuitry, such as circuitry 126. Alternatively, PMU 204 and performance monitor driver 208 may individually or together be embodied as

hardware and/or firmware in circuitry, such as circuitry 126. Memory 206 and event buffer ~~204~~ 218 may each comprise memory such as memory 104, 128. For example, memory 206 may be one or more registers of a processor, such as processor 202, and event buffer ~~204~~ 218 may be a temporary memory.

Please replace paragraph [0036] with the following amended paragraph:

[0036] FIG. 4 is a flowchart illustrating a method according to another embodiment. The method begins at block 400, and continues to block 402 where one or more client applications 212 may read one or more processed event records 216 from event buffer ~~204~~ 218, each processed event record 216 including one or more processed event data corresponding to one or more unprocessed event data. At block 404, client application 212 may generate client uncompressed event data 220 corresponding to the one or more uncompressed event data. Generating one or more client uncompressed event data may include outputting an event datum if the event datum is not in a compressed format 404A, or decompressing an event datum if the event datum is in a compressed format 404B. The method ends at block 406. Client application 212 may use decompressor 214 to decompress compressed event data. Decompressor 214 may be local to each client application 212, where each client application 212 may have its own decompressor 214. Alternatively, decompressor 214 may be global with respect to each client application 212, where decompressor 214 may decompress event data for the one or more client

applications 212.

Please replace paragraph [0094] with the following amended paragraph:

[0094] Client application 212 may read one or more event records from event buffer 204 218. In one embodiment, performance monitor driver 208 may notify client application 212 when event buffer 204 218 is full. Client application 212 may read each event record in event buffer 204 218. For each event datum, if the event datum is compressed, decompressor 214 may decompress the event datum to generate client uncompressed event datum 220. If the event datum is not compressed, the event datum may be used as the client uncompressed event datum 220. event buffer 204 218 may maintain information about address dictionary, such as the size, organization, and hash function used. Decompressor 214 may maintain a copy of address dictionary 910, 1010 separately from compressor 210.